

Unidad 4- Arreglos, diccionarios y manipulación de datos

¿Qué es un arreglo?

Un arreglo, también conocido como array en inglés, es una estructura de datos que almacena una colección de elementos del mismo tipo. Los elementos se organizan en posiciones numeradas llamadas índices, que comienzan desde 0 en adelante. Esto permite acceder a cada elemento mediante su índice.

Por ejemplo, en un arreglo de números enteros, podríamos tener un arreglo llamado "números" que contiene los valores "{1, 2, 3, 4, 5}". Para acceder al valor '3', que está en la tercera posición del arreglo, usamos 'números[2]', ya que los índices comienzan en 0.

Los arreglos son muy útiles para trabajar con conjuntos de datos relacionados que necesitan ser procesados de manera conjunta, como una lista de nombres, valores numéricos, etc.

¿Cómo declarar un arreglo en PHP?

Para declarar un arreglo en PHP debemos realizarlo por cualquiera de las siguientes maneras

```
<?php
$nombrres = array("Thor", "Loki", "Odin");
```

```
<?php
$nombrres= ["Thor", "Loki", "Odin"];
```

En PHP hay 3 tipos de array:

- 1) **Arreglos con índice**- Son arreglos con índice numérico (como en la definición)
- 2) **Arreglos asociativos** - Son arreglos con claves asociadas
- 3) **Arreglos multidimensionales**- Son arreglos que pueden contener uno o más arreglos

A diferencia de otros lenguajes de programación, yo puedo guardar cualquier tipo de dato dentro de un arreglo (incluso datos que no posean el mismo tipo como por ejemplo un boolean false, un número 3 y un string "Hola", todo en un mismo arreglo)

Arreglos indexados

Los arreglos indexados son aquellos a los que puedo acceder de manera directa utilizando su número de posición

```
<?php
    $nombres = array("Thor", "Loki", "Odin");

    $nombres[0] = "Ragnarok";

    echo $nombres[0];
```

En el ejemplo anterior, se asigna en base a su posición, el nombre de la posición 0, modificándolo por el nuevo valor.

Si queremos mostrar todos los elementos del arreglo, podemos recorrerlo con la estructura de control foreach (en caso de que esté ordenado).

```
<?php
    $nombres = array("Thor", "Loki", "Odin");

    foreach ($nombres as $i) {
        echo "$i <br>";
    }
```

En PHP, los arreglos no son estáticos, así que podemos agregar nuevos valores al arreglo, indicando su índice correspondiente, inclusive si ponemos índices no correlativos. Además, una vez que asignamos un valor a un arreglo, el mismo se crea, así que no es necesario declararlo en un inicio.

```
<?php
    $nombres[0] = "Thor";
    $nombres[4] = "Loki";
    $nombres[8] = "Odin";

    array_push($nombres, "Fenrir");

    print_r($nombres);
```

La función `array_push()` me permitirá agregar un nuevo elemento al final del arreglo y el mismo poseerá el índice del mayor índice ingresado +1

Array ([0] => Thor [4] => Loki [8] => Odin [9] => Fenrir)

Arreglos asociativos

Los arreglos asociativos asignan una clave para cada valor que poseen (veremos ejemplos de los arrays asociativos mas importantes mas adelante)

```
<?php
    $persona = array("nombre"=>"Christian", "apellido"=>"Barrios", "edad"=>26);

    echo $persona['edad'];
```

En el ejemplo anterior, se muestra el dato asociado a la clave "edad" dentro del arreglo. Se puede modificar el dato de un arreglo haciendo referencia a su clave

```
<?php
    $persona = array("nombre"=>"Christian", "apellido"=>"Barrios", "edad"=>26);

    $persona['edad'] = 24;
    echo $persona['edad'];
```

Podemos desplegar la información de todo el arreglo utilizando la estructura `foreach` también

```
<?php
    $persona = array("nombre"=>"Christian", "apellido"=>"Barrios", "edad"=>26);

    foreach ($persona as $i => $j) {
        echo "$i: $j <br>";
    }
}
```

En el caso anterior, la variable \$i representará todas las claves y la \$j todos los valores asociados.

Arreglos multidimensionales

Los arreglos multidimensionales contienen arreglos dentro de su estructura (puede ser uno o más de uno y estos pueden ser asociativos, indexados u otros multidimensionales)

```
<?php
    $personas = array (
        array("Juan","Perez",18),
        array("Gustavo","Trelles",23),
        array("Karina","Rivas",41),
        array("Sandra","Martinez",32)
    );

    for ($fila = 0; $fila < 4; $fila++) {
        echo "<p><b>Numero de fila $fila</b></p>";
        echo "<ul>";
        for ($columna = 0; $columna < 3; $columna++) {
            echo "<li>".$personas[$fila][$columna]."</li>";
        }
        echo "</ul>";
    }
}
```

En el ejemplo anterior se define un arreglo de personas y se muestran con una estructura de control llamada "for", los datos de cada una de las personas.

Existen arreglos especiales los cuales podremos utilizar para manipular la información del servidor en el que se aloje este sitio web (para mas referencia vea el capítulo 2 y 3 del libro "Learn PHP que se encuentra en este mismo repositorio")

Manipulación de datos

Para realizar la manipulación de datos dependemos de aplicar los formularios vistos en el apartado de frontend (Diseño Web II)

Crearemos un formulario con html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulario</title>
</head>
<body>
  <form action="ejemploResp.php" method="get">
    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre" name="nombre"><br>
    <label for="apellido">Apellido:</label><br>
    <input type="text" id="apellido" name="apellido">
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

En este formulario ingresamos un nombre y un apellido, cada uno en un campo de texto.

A continuación, definimos nuestro botón de “submit” En el cabezal del “form” debemos indicar dos términos: primero action (página de destino de los datos que acabamos de ingresar) y segundo method (tipo de envío de datos los cuales pueden ser GET o POST)

En el caso del método de envío, la gran diferencia entre GET y POST es la visibilidad de los datos enviados. En el caso de GET, los datos se envían a través de un apartado asociado a la URL a partir de la referencia “?”. Los datos son completamente visibles en la URL al enviarlos. En el caso de POST, los datos se envían a parte de la URL, de manera oculta. Este último método es el utilizado por defecto y es el requerido en caso de enviar información sensible.

Veamos el contenido de “ejemploResp.php”

```
<?php
    $nombre=$_GET["nombre"];
    $apellido=$_GET["apellido"];

    echo "Hola $nombre $apellido";
```

En este caso el arreglo asociativo que llamo es el \$GET pero existen otros que nos pueden servir dependiendo de nuestro caso concreto (\$_POST, \$_REQUEST, \$_SERVER, \$_COOKIE, \$_SESSION, \$_FILES, etc)